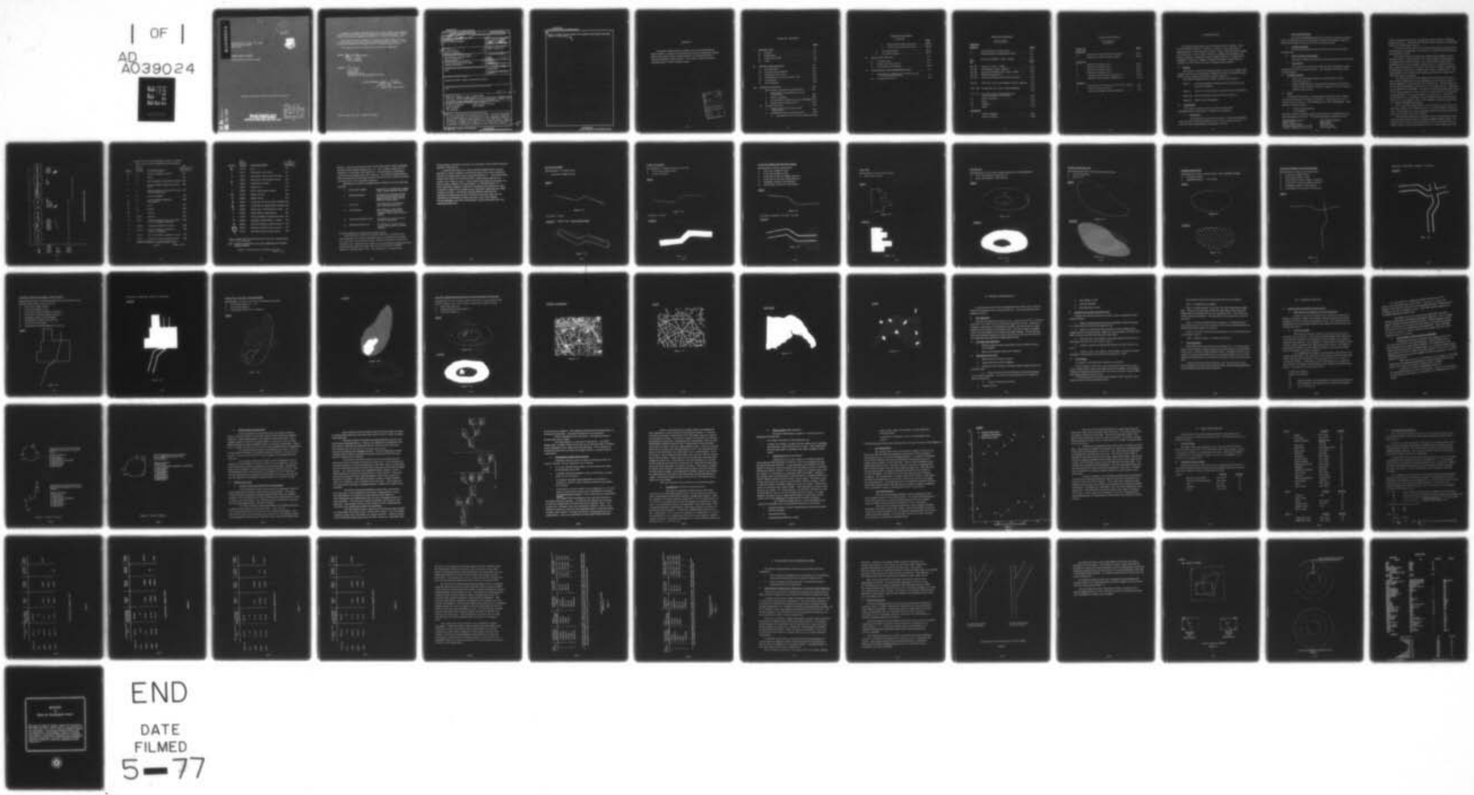
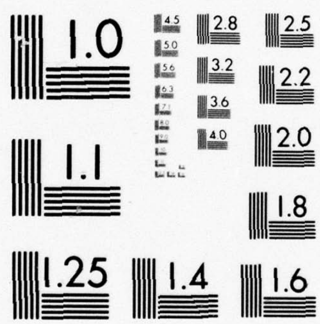
AD-A039 024 PRC INFORMATION SCIENCES CO MCLEAN VA F/G 9/2
RASTER IMAGING SOFTWARE.(U)
MAR 77 R K LUBBES, P A MACERA, K S PRZEWLOCKI F30602-74-C-0345
UNCLASSIFIED RADC-TR-76-340-VOL-1 NL

1 OF 1
AD
A039024



END
DATE
FILMED
5-77



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

(12)

NW

RADC-TR-76-340, Volume I (of three)
Final Technical Report
March 1977

RASTER IMAGING SOFTWARE

PRC Information Sciences Company

Approved for public release; distribution unlimited.

ROME AIR DEVELOPMENT CENTER
AIR FORCE SYSTEMS COMMAND
GRIFFISS AIR FORCE BASE, NEW YORK 13441

Volume III, "Computer Documentation" has not been placed in the DDC/NTIS collection because of its limited usefulness to those outside RADC. Volumes I and II will provide adequate information for the user or potential user.

This report has been reviewed and is approved for publication.

APPROVED: John R. Baumann
JOHN R. BAUMANN
Project Engineer

APPROVED: H Davis
HOWARD DAVIS
Technical Director
Intelligence and Reconnaissance Division

FOR THE COMMANDER: John P. Huss
JOHN P. HUSS
Acting Chief, Plans Office

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
RADC-TR-76-340-Volume I (of three)

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
RASTER IMAGING SOFTWARE

5. TYPE OF REPORT & PERIOD COVERED
Final Technical Report
July 1974 - June 1976

6. PERFORMING ORG. REPORT NUMBER
N/A

7. AUTHOR(s)
R. Kirk Lubbes
Patricia A. Macera
Kathryn S. Przewlocki

8. CONTRACT OR GRANT NUMBER(s)
F30602-74-C-0345

9. PERFORMING ORGANIZATION NAME AND ADDRESS
PRC Information Sciences Company
7600 Old Springhouse Rd
McLean VA 22101

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
63701B
32030301

11. CONTROLLING OFFICE NAME AND ADDRESS
Rome Air Development Center (IRRP)
Griffiss AFB NY 13441

12. REPORT DATE
March 1977

13. NUMBER OF PAGES
74

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)
Same

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)
Same

407 126

18. SUPPLEMENTARY NOTES
RADC Project Engineer: John R. Baumann (IRRP)
Volume III, "Computer Documentation" has not been placed in the DDC/NTIS collection because of its limited usefulness to those outside RADC. Volumes I and II will provide adequate information for the user or potential user.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
Automated Cartography
Raster Processing
Raster Plotters
Area Symbology
Lineal to Raster Conversion
Type Placement

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The Raster Imaging Software was developed to support raster plotting capabilities existing at RADC. The software consists of (1) interactive symbol and text placement capability which operates on the RADC Experimental Compilation Console; and (2) batch processing software to convert lineal data to raster for plotting. Included in the lineal to raster conversion software in addition to creating line weights is the capability to fill areas with a solid fill or pattern fill, to fill between contours of different elevations (elevation

over

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

tints) to perform priority masking, and to generate point symbols and alpha-numerics in raster format.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

ABSTRACT

The Raster Imaging Software (RIS) system developed under an RADC contract, enhances the existing Experimental Compilation Console (ECC) and provides raster imaging support to the existing Format Conversion Software (FCS) System. This volume contains the Raster Imaging Software Final Technical Report.

TABLE OF CONTENTS

TABLE OF CONTENTS

(continued)

TABLE OF CONTENTS

List of Figures

TABLE OF CONTENTS

List of Figures
(continued)

I. INTRODUCTION

The Raster Image Software is documented in three volumes. Final Technical Report, Volume I, contains a summary of the entire Raster Imaging Software effort including analysis, design, implementation, testing, and conclusions. User's Guide, Volume II, contains the operational procedures required to exploit the Raster Imaging Software. Program Documentation, Volume III, contains a detailed description of the programs modified or written under the Raster Imaging Software effort.

A. Purpose

The purpose of the Raster Imaging Software effort was to enhance the current research and development compilation system and extend the capabilities of the existing raster plotter support software.

The Raster Imaging Software effort consisted of four phases:

Phase I	Test and Evaluation of Existing ECC/CDP and Format Conversion Software.
Phase II	Format Conversion Software Improvement and Extension
Phase III	ECC/CDP System Improvement and Extension
Phase IV	Final Test and Evaluation

B. Organization

The Final Technical Report, Volume I of the Raster Imaging Software Final Report, consists of six sections:

1. Introduction

The Introduction describes the purpose of the Raster Imaging Software and describes the structure of the raster imaging services under the RADC Experimental Cartographic Facility.

2. <u>General Methodology</u>

The General Methodology section describes the test and evaluation of the original Experimental Cartographic Console/Cartographic Digitizing Plotter (ECC/CDP) and Format Conversion Software (FCS).

3. <u>Technical Results</u>

The Technical Results section describes the system structure and data flow.

4. <u>Final Testing and Evaluation</u>

The Final Testing and Evaluation Section documents the test results.

5. <u>Conclusions</u>

The Conclusions section describes the PRC viewpoint on further development of raster imaging services. The limitation of the RIS system are also defined.

C. <u>Technical Problem</u>

- o Study, test and evaluate RADC's current lineal to raster conversion process.
- o Write, implement and test and updated expanded and optimized lineal to raster software system to support current and expected hardware data formats.

D. <u>Scope</u>

The ECC/CDP System is an experimental test bed for the evaluation of automated interactive compilation functions. These functions may be divided into two classes: 1) Data Selection and 2) Data Manipulation. Data Selection functions include:

- o Feature Class Selection
- o Area Selection

These two criteria define the graphic data set which is manipulated by the Data Manipulation Functions. The Data Manipulation Functions include:

Display/Modify Header	Join Feature Segments
Delete Feature	Select Scale
Insert Contour Label	Move Window
Position Point Symbols and Text	Position Cursor
Transfer Graphic Cursor to CDP	Write Output Tape

These ten functions provide the cartographer with a means to modify the graphic data base. The output from the ECC/CDP system is a cartographic feature file in MMS lineal format.

The Insert Contour Label, and Position Point Symbol and Text, ECC/ CDP functions were specified as functions to be improved under the Raster Imaging Software effort.

The Insert Contour Label, and the Position Point Symbol and Text functions produce a pseudo-MMS record which defines a point symbol or an alphanumeric character string. The Pseudo-MMS record defines a graphic element or set of graphic elements which is comprised of more than a single straight line. The basic structure is, however, identical to the MMS record format which defines only a single continuous line. The pseudo-MMS record contains only alphanumeric data and a single line to define the symbol orientation.

The Point Symbols and Text positioned by the ECC/CDP system require raster imaging prior to their plotting on the final names overlay. Raster imaging is a two-step process (see Figure 1). The first step is to convert the pseudo-MMS record to one or more MMS records. This operation is performed by PSUMMS. All logical relationships between the characters of a given character string or parts of a point symbol are eliminated. A separate MMS record is generated for each character. Therefore, a pseudo-MMS record containing a string of six ASCII characters will generate six or more MMS records, each containing a single continuous line as a result of PSUMMS processing.

The second step of the process is the conversion of the center line character casings to area filled characters or point symbols in raster format. This operation is a raster imaging function and is therefore performed by the Format Conversion Software.

The Format Conversion is the prime area of concentration for improvement and extension under the Raster Imaging Software effort. The purpose of the Format Conversion Software is to convert lineal files in MMS format to raster formatted files in order to drive high speed raster

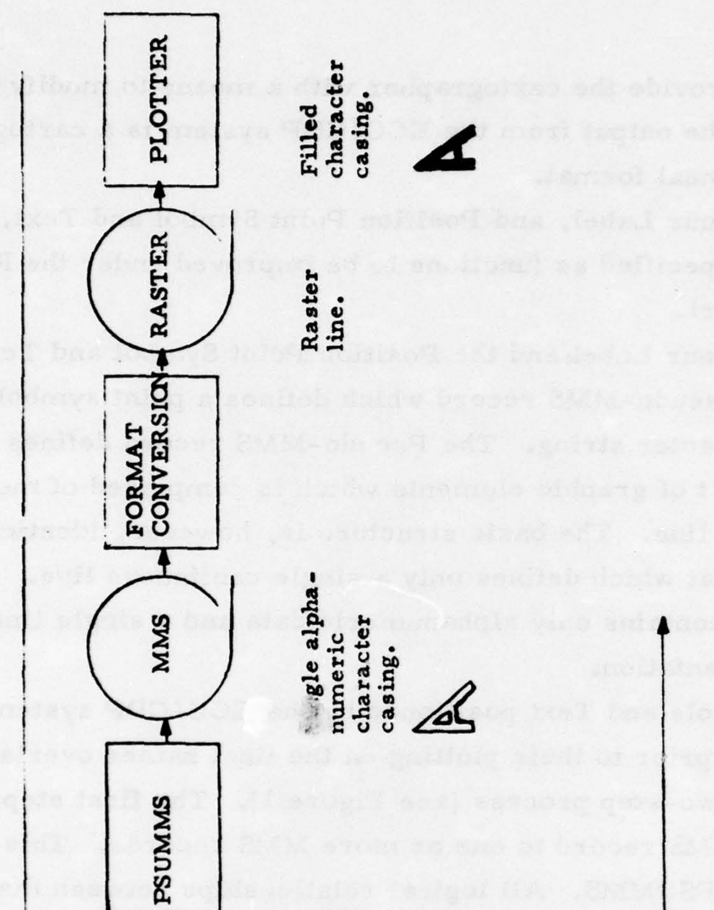

System Flow
ECC/CDP
PSEUDO-MMS
PSUMMS
MMS
FORMAT CONVERSION
RASTER
PLOTTER
Data Format of text per record
Alphanumeric string in ASCII and base line.
Single alpha numeric character casing.
Raster line.
Filled character casing.
Plot of data in file
PDP-9/ PDP-15 HIS-635
TEXT GENERATION AND DATA FLOW

Figure 1

DEFINITION OF F-CODE (Point Symbol Numbers) FOR POINT SYMBOL PSEUDO-MMS RECORD

Symbol	Point Symbol Number	Name/Description	Jog Specification Number
	1	Spring, Fountain, Cistern	344
	2	Well	345
	3	Spot elevation, unidentifiable point	407
	3	Spot elevation, highest in general area	408
	3	Spot elevation, questionable value, unidentifiable point	410
	3	Tank (all types)	732
	4	Spot elevation, highest on compilation	408
	5	Cave	447
	6	School	612
	7	Church	613
	8(10)	Mosque	613
	9(11)	Small developed areas, less than .15" at narrowest dimension	602
	10(12)	Landmark features or object	609
	11(13)	Point of change in number of tracks or of gauge	712
	12(14)	Railroad station, position unknown	713
	13(15)	Railroad station, position known	713

POINT SYMBOLS & ALPHANUMERICS KEY

Figure 2 (Page 1 of 2)

Symbol	Point Symbol Number	Name/Description	Jog Specification Number
⚒	14(16)	Mine	729
o	15(17)	Well (Other than water)	731
△	16(20)	Horizontal control point (triangle)	750
•	18(21)	*Horizontal control point (circle)	750
⊕	18(22)	Astronomic position	751
+	19(23)	Sunken rock	910
*	20(24)	Rock, uncovering or awash	911
	21(25)	Exposed wreck	913
	22(26)	Sunken wreck	914
	23(27)	Anchorage for boats (small vessels)	929
⚓	24(30)	Anchorage for ships (large vessels)	930
□	25(31)	Small areas of hutments or kraals	611
|	26(32)	Road location, approximate	816
|	27(33)	Point of change in road information	823
o	28(34)	Aqueduct, tunnel shaft, or outlet	326
■	29(35)	Road interchange, pattern unknown	814
	30(36)	National (Federal) route markers	826
o	31(37)	Secondary (state) route markers	827

*This symbol exists in the library but is entered automatically, never by the operator.

Note: numbers in brakets are the octal equivalents for the point symbol numbers.

Figure 2 - Point Symbols & Alphanumeric Key
(Page 2 of 2)

plotters. The first generation of the FCS provided only a means of plotting the feature data without enhancement, i. e., the resulting raster plot was identical to a lineal plot of the input data. The advantages of obtaining a raster plot are plotting speeds and accuracy. The Raster Imaging Software effort was directed toward enhancement of the lineal data to produce finished raster plots for color separation negatives.

Specifically, the Raster Imaging Software (RIS) provides the following functions:

1.	Areal Line Weights	Generation of variable line weights using a single specified spot size.
2.	Priority Masking	Assignment of priorities to feature classes such that only the highest priority feature will be plotted at intersections.
3.	Area Fill	The filling of any closed area with a specified spot size.
4.	Fill Between	The filling of a closed band which is defined by a feature contained wholly within a second feature.
5.	Screen and Hatchure Fill	The filling of a closed area by screens or hatchures.
6.	Repeated Symbol Fill	The filling of a closed area by a given symbol pattern cyclicly repeated.

The above patterns are generated in raster format.

These functions which are provided to the user by the RIS can be used separately or in groups depending upon the results that are needed to be achieved in the final raster plot. Pictorial examples of each of the functions, and a representative sample of the permutations of these functions are provided to show the system user some of the options available (Figure 1A through Figure 11B). The instructions to construct the

various options available to the user are contained in User's Guide Volume II Appendix D Job Stream.

A typical task would be to extract specific features from a source document in order to produce a less dense subset of the source document. The user is working with a source document (Figure 12) where the roads, contours and cities are of prime interest, each of these features are needed to be a different color. Three raster plots should be produced; one each of roads, contours and cities (Figure 13 through Figure 15). Color separation negatives would then be produced and overlayed in order to produce a subset of the original source document containing only roads, contours and cities. The Raster Imaging Software effort has the capability of performing the above task plus many other tasks that are generated by combining its various functions Areal Line Weights, Priority Masking, Area Fill, Fill Between, Screen and Hature Fill, and Repeated Symbol Fill.

Areal Line Weights

Specifications: To Thicken Line

1. Areal Line Weight 20 mils

INPUT

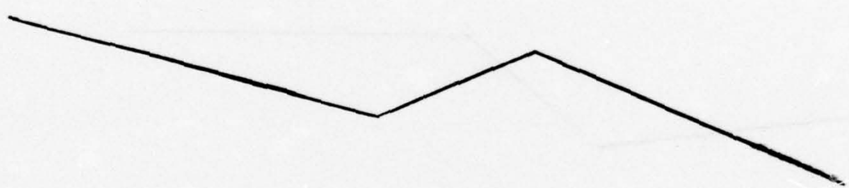


Figure 1A

let $\frac{1}{2}$ inch = 20 mils

OUTPUT CASED LINE (LFEC Module Only)

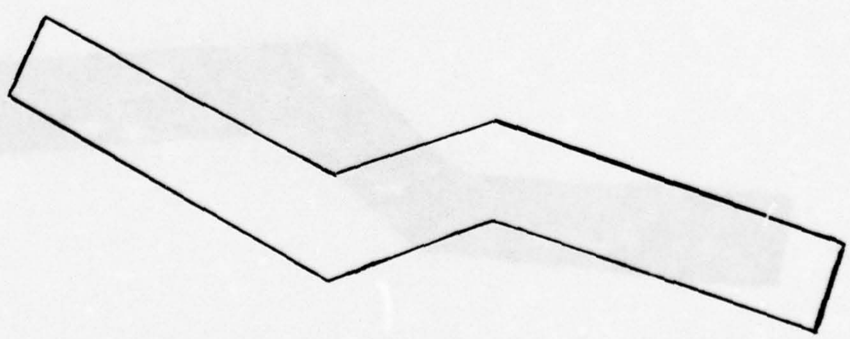


Figure 1B

Areal Line Weights

Specifications To Thicken Line and Areal Fill

1. Areal Line Weight 20 mils
2. Areal Fill Output

INPUT

Figure 2A

let $\frac{1}{2}$ inch = 20 mils

OUTPUT

Figure 2B

Areal Line Weights and Priority Masking

Specifications To Form Cased Roads

1. Areal Line Weight 20 mils
2. Areal Line Weight 15 mils
3. Areal Fill 20 mil Line Output Channel 3
4. Areal Fill 15 mil Line Output Channel 2
5. Disengage channel 2 when plotting
6. Set priority 15 mil line $>$ 20 mil line

INPUT

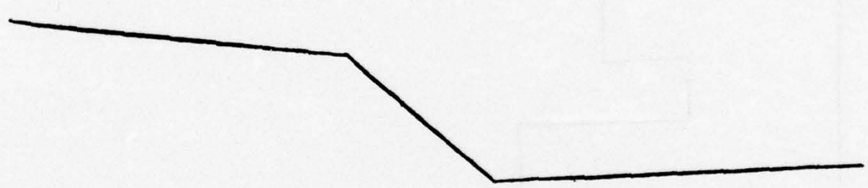


Figure 3A

Let $\frac{1}{2}$ inch = 20 mils, 6/16 inch = 15 mils

OUTPUT

Figure 3B

Area Fill

Specifications To Fill Closed Feature

1. Area Fill Feature

INPUT

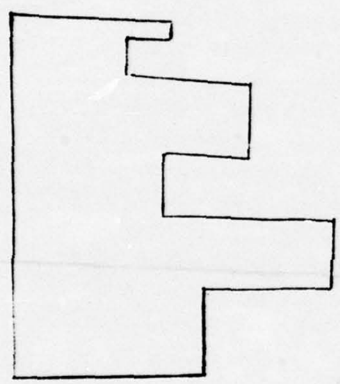


Figure 4A

OUTPUT

Figure 4B

Fill Between

Specifications To Fill Feature wholly enclosed in a second feature.

1. Area Fill Feature A with color #1
2. Area Fill Feature B with Color #2

INPUT

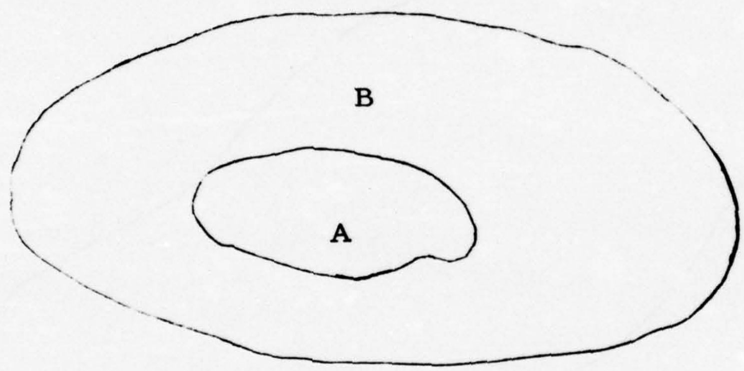


Figure 5A

OUTPUT

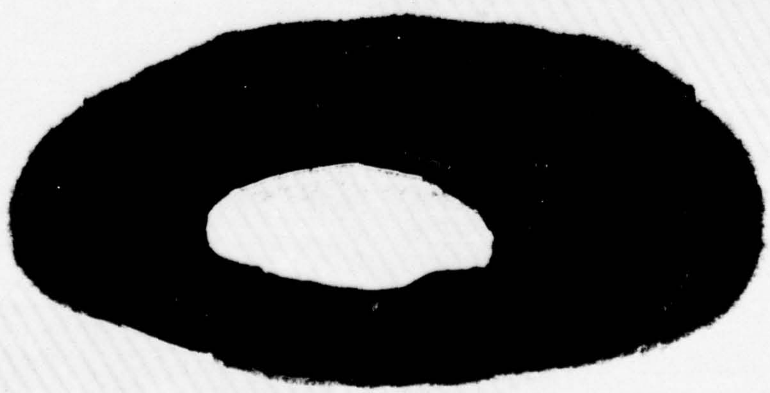


Figure 5B

Screen and Hatchure Fill

Specifications To Fill a Closed Feature with Screen

1. Area Fill Feature
2. Screen 45^{o}

INPUT

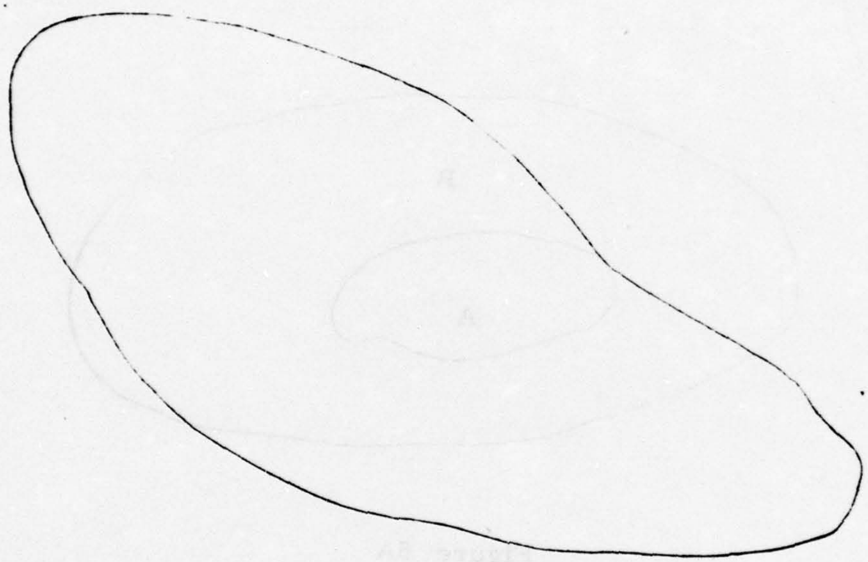


Figure 6A

OUTPUT

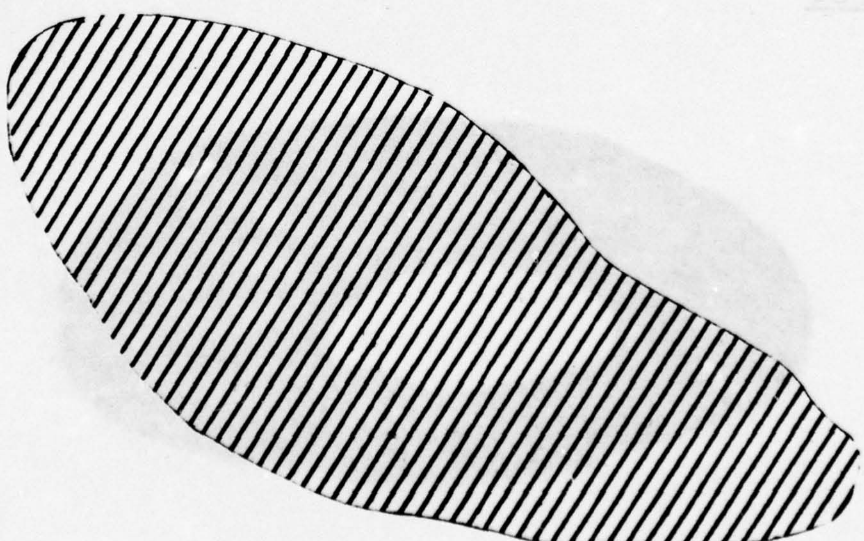


Figure 6B

Repetitive Symbol Fill

Specifications To Fill a Closed Feature with a Repetitive Symbol

1. **Area Fill Feature**
2. **Repetitive Symbol** **Rice Pattie**

INPUT

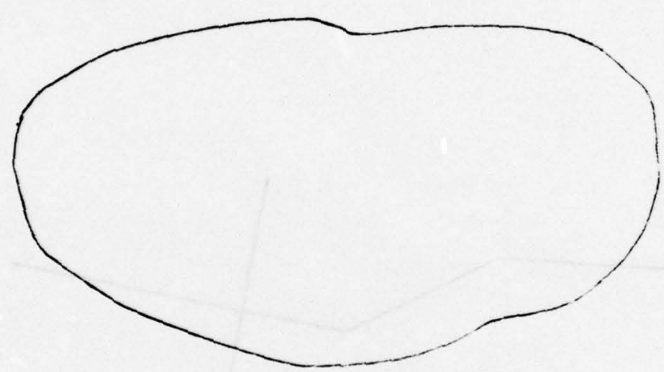


Figure 7A

OUTPUT

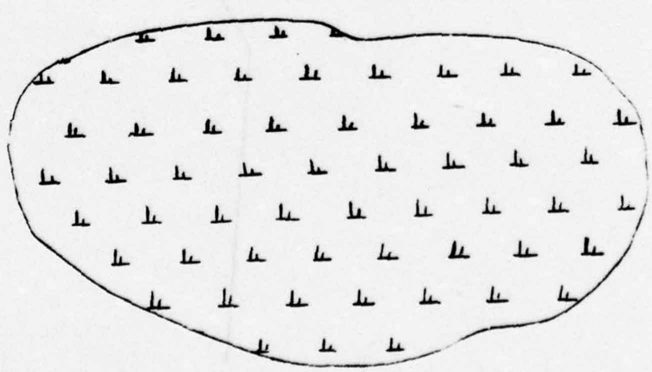


Figure 7B

Areal Line Weights and Priority Masking

Specifications To Form Cased Intersection

1. Areal Line Weight 20 mils
2. Areal Line Weight 15 mils
3. Areal Fill 20 mil line output to channel 3
4. Areal Fill 15 mil line output to channel 2
5. Disengage channel 2 when plotting
6. Set priority of 15 mil line > 20 mil line

INPUT

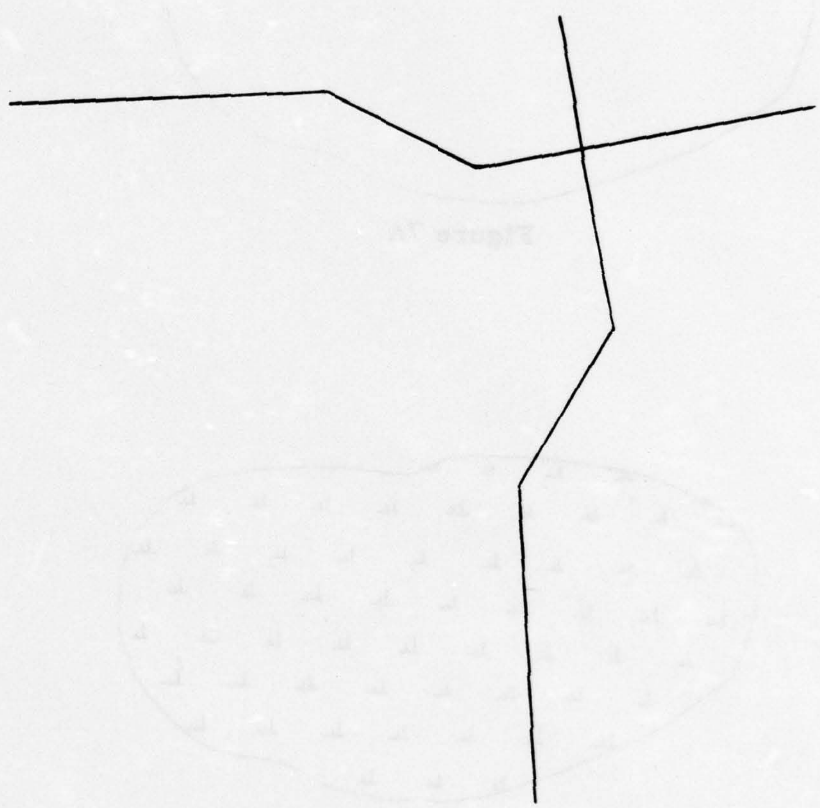


Figure 8A

let $\frac{1}{2}$ inch = 20 mil line, 3/8 inch = 15 mil line

<u>OUTPUT</u>

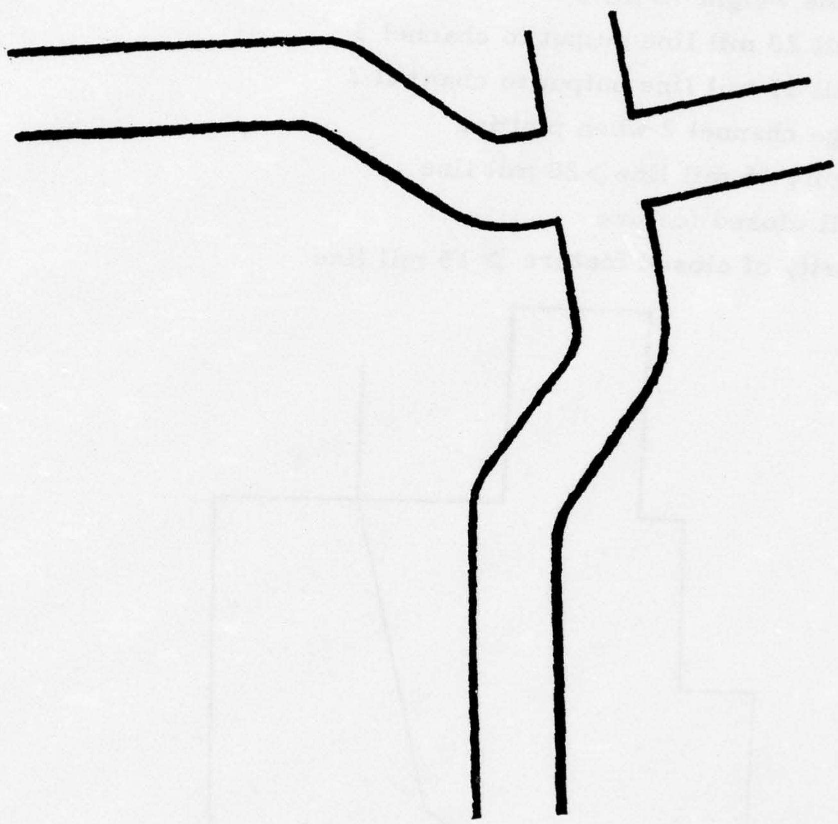


Figure 8B

Area Fill, Areal Line Weights, Priority Masking

Specifications To form cased intersection with closed feature where priority closed feature $>$ cased feature.

1. Areal Line weight 20 mils
2. Areal Line weight 15 mils
3. Areal Fill 20 mil line output to channel 3
4. Areal Fill 15 mil line output to channel 2
5. Disengage channel 2 when plotting
6. Set priority 15 mil line $>$ 20 mil line
7. Area Fill closed feature
8. Set priority of closed feature $>$ 15 mil line

INPUT

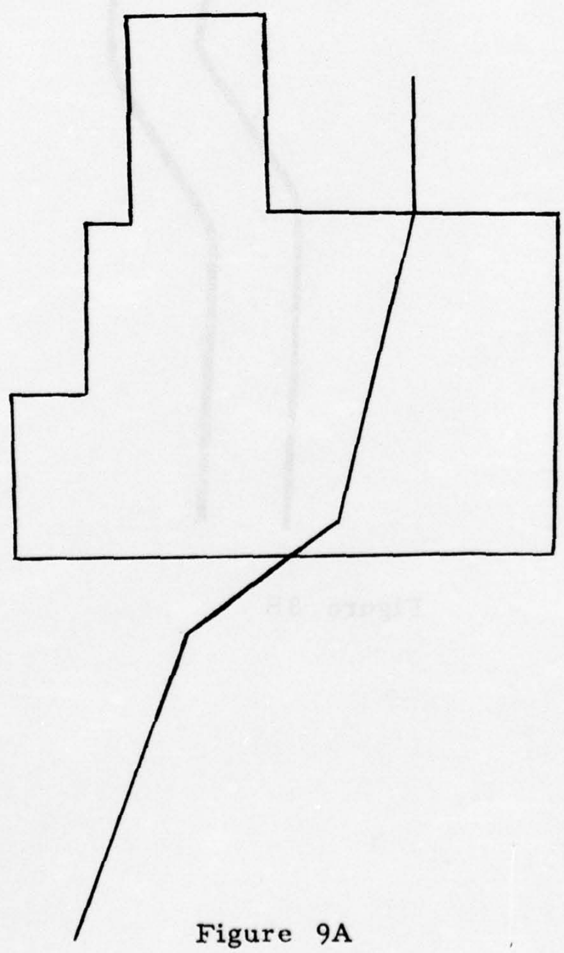


Figure 9A

let $\frac{1}{2}$ inch = 20 mil line, 3/8 inch = 15 mil line.

OUTPUT

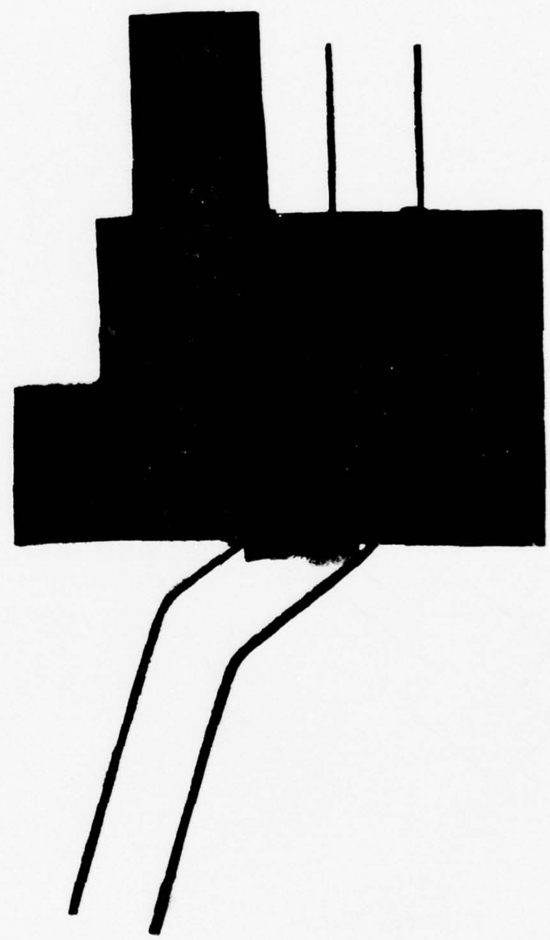


Figure 9B

Screen Fill, Area Fill, Priority Masking

Specifications To form Screen Fill, overlayed by Area Fill

1. Screen Fill Feature A 135^{o}
2. Area Fill Feature B
3. Set Priority Feature B > Feature A

INPUT

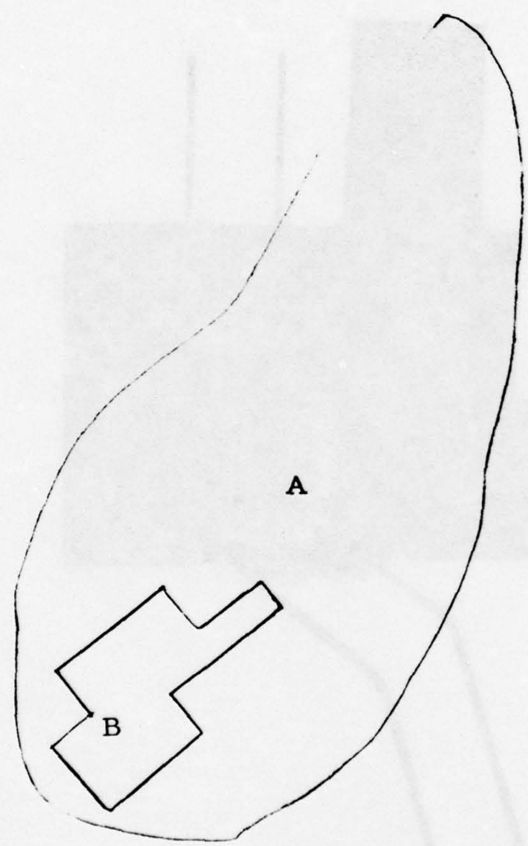


Figure 10A

OUTPUT

Figure 10B

Area Fill, Repeated Symbology Fill, Priority Masking, Fill Between

Specifications To fill a Feature wholly enclosed in a second feature with repeated symbol, and to fill a feature wholly inside another feature.

1. Area Fill Feature A
2. Repeated Symbol Fill Feature B
3. Area Fill Feature C

INPUT

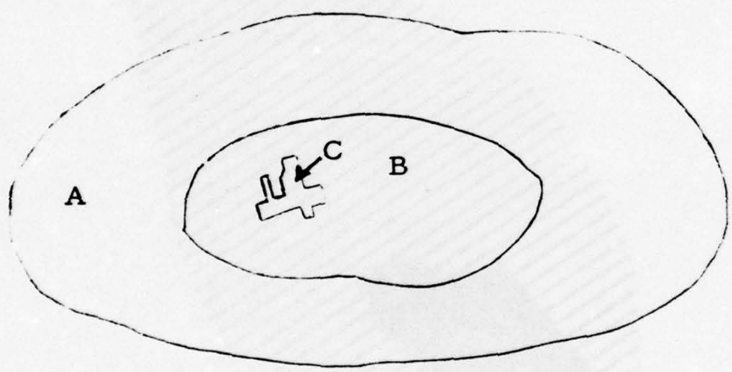


Figure 11A

OUTPUT

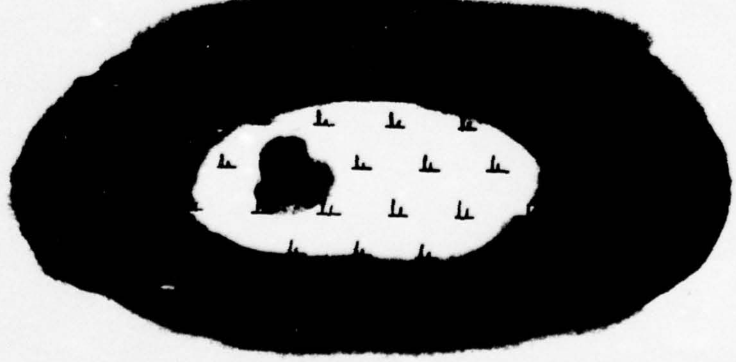


Figure 11B

SOURCE DOCUMENT

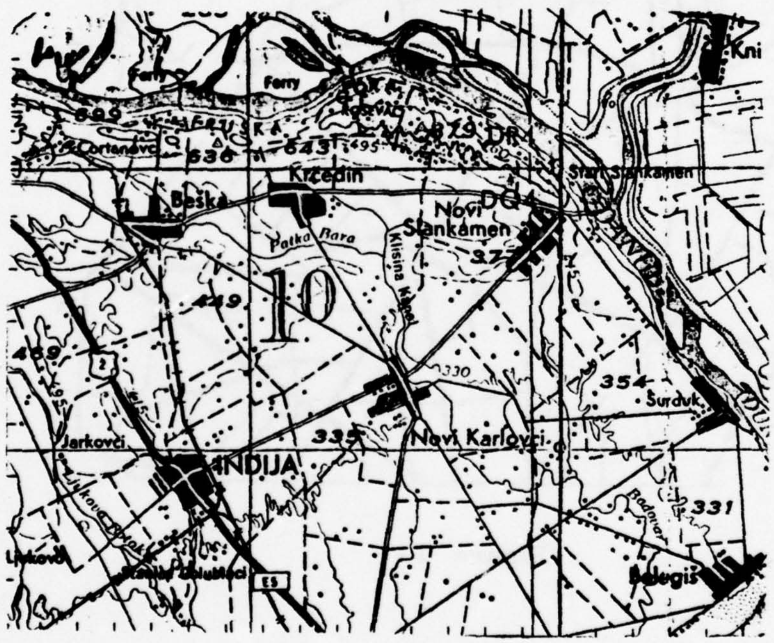


Figure 12

ROADS

Figure 13

CONTOURS

Figure 14

CITIES

Figure 15

II. GENERAL METHODOLOGY

The initial phase of the Raster Imaging Software effort was to test and evaluate the existing raster conversion process. The test procedures were defined as follows.

A. Test Data Base

The NL 34-8 JOG Sector defined by the coordinate (N 45° 00', E 20° 00') and (N 45° 12', E 20° 36') was used as the initial test data set. This sector contains a relatively even distribution of graphic data and contains a representative cross section of cartographic data. Only the contours, drainage, roads, and city outlines were required for initial testing. The section and extraction functions required to extract this data set from the NL 34-8 JOG were executed from Batch Processing System (BPS).

B. Test Data Base Statistics

- o Feature counts by class and number inches of data per class were obtained.
- o Plots of each feature class were obtained.

C. Preliminary ECC Test

1. Input process all feature classes.
2. Inspect plot file via CRT displays.
3. Execute all ECC functions carefully noting response times and problem areas.

a. Name Placement and Point Symbol Placement functions were carefully evaluated. All characters and symbols were verified as names were placed.

b. Contour Labeling was tested.

4. Output Process

5. Plot Output on CDP

6. Execute PSUMMS

7. Plot final tape on CDP

D. Preliminary Format Conversion Tests

Perform the following Format Conversion runs by coding the control cards as follows:

1. Extract and plot center line plots specifying a different channel (i.e., line out) for each different feature class.

2. Extract roads and specify areal fills for three different line weights relative to each subclass. Make two runs specifying 1000 lines/inch with 1 mil spot, and 500 lines/inch with 2 mil spots.

3. Extract lakes, city outlines, and pseudo-MMS codes (R=33,35) and specify area fills on each of these classes.

4. Extract contours and specify fill between two sets of index contours.

5. Extract roads, city outlines, and drainage, and specify priority masking for city outlines over roads and drainage over roads.

E. Test Results

A summary of the initial test results are as follows:

The Preliminary ECC Tests were successful in all functions except Insert Contour Label, where the contour label did not appear on the updated display and Position Point Symbol and Text, where character spacing within text strings generated was in error.

The Preliminary Format Conversion Software tests, however, indicated a number of serious problem areas.

The results of the Format Conversion tests were as follows:

Test 1 - Completed successfully.

Test 2 - System aborted on the first run while producing an output tape. The plot of the partially completed output tape showed areal line weight with no areal fills and also two extraneous lines. The second run reached normal termination. The plot produced had numerous errors in areal fill.

Test 3 - The job reached normal termination. The plot produced showed numerous errors in the area fills such as lines connecting features and holes in the fill portion of a feature.

Test 4 - The job reached normal termination, but the plot tape was in an unreadable format.

Test 5 - System aborted. No plot was produced.

F. Test Evaluation

ECC/CDP system performed most functions well. A major problem exists in the Contour Labeling function which would not plot a defined label. There were a number of minor problems noted. Specifically the errors in character spacing within alphanumeric text strings defined under the Point Symbol and Alphanumeric Text Position Function are particularly objectionable.

Format Conversion testing indicated many major problem areas. Errors existed in the areal line weight function, priority masking function, area fill function and fill between functions.

III. TECHNICAL RESULTS

A. Experimental Compilation Console (ECC)

1. Point Symbol and Alphanumeric Text Positioning

Problem areas were encountered in scaling and relative positioning of characters within a text string. Analysis showed both of these problems were due to minor programming errors. The Point Symbol and Alphanumeric Text Positioning software was modified to improve the man-machine interaction.

2. Contour Labeling

One function of the ECC system is the generation of contour elevation numeric strings. The result is properly plotted interrupted contours with the elevations correctly spaced, right-reading, and conformal within the interruptions. This clipping is accomplished by building a buffer for each contour containing the clips for the given contour. The buffer is passed from the PDP-15 to the PDP-9 through the interprocessor buffer where the plot file is properly modified.

The clipping process begins as the baseline orientation (for the contour label)is input. Both end points of the baseline must be on the same feature and should be designed so the baseline will follow the contour as closely as possible. A clip buffer is built defining the two x, y points on the contour to be clipped; the label being on the straight line between those points. The buffer has the following format:

header block address
header word address
x_1
y_1 Where the block and word address references the first word
x_2 of that feature in the plot file and the clip occurs between
y_2 x_1, y_1, and x_2, y_2.
⋮

As many buffers as required are filled (42 clips each) and stored on the Burrough's disc on the PDP-9. The first x, y pair is the highest x value for the character and corresponding baseline y determined for plotting on the CRT for operator review. Up to 256 separate contours may be labeled with up to 20 labels each.

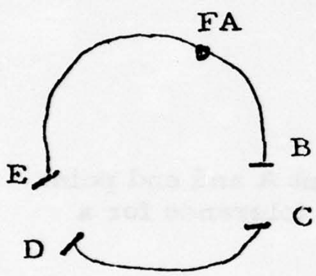
The Output Processor writes pseudo-MMS header and data records immediately preceeding the appropriate feature(s). When a clipped contour is found, subroutine OPCLIP is called. OPCLIP outputs a pseudo header, then builds and outputs the pseudo header, then builds and outputs the pseudo data record(s) defining the clip segment(s). Upon completing all clips for a feature, control is returned to the Output Processor which outputs the feature as normal.

3. Pseudo-MMS to MMS Conversion (PSUMMS)

PSUMMS provides the link between the output processor and the input to the Format Conversion Software. Pseudo-MMS records, which contain only the point symbol identifier or the alphanumeric text string and their associated orientation angle, are converted to MMS format which defines the lines to outline the graphic symbol. The lack of a working PSUMMS program prevented the plotting of point and alphanumeric symbols output from the ECC.

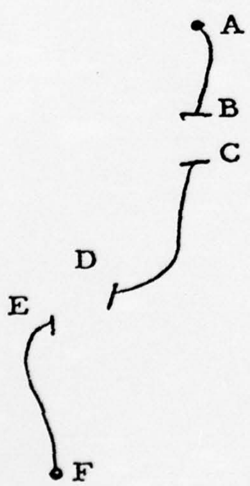
Numerous errors were encountered in the debugging of PSUMMS. Initial plots of the point and alphanumeric symbols pointed out a transformation error in the ECC software which was then rectified.

Routines were added to PSUMMS to allow the Pseudo-MMS and MMS records on the input file. Thus new features may be added after the generation of a point symbol or alphanumeric character string.



Begin point A and end point F are within the tolerance for a closed feature.

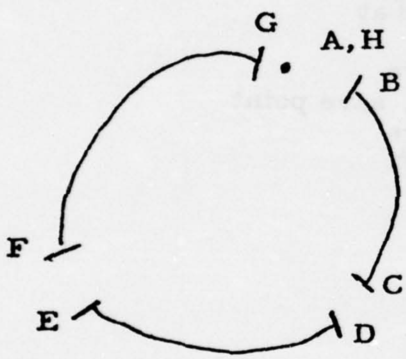
Begin at A and at
B save point
C begin feature
D end feature, save point
E begin feature
F get data at A
B end feature



Begin point A and end point F are beyond the tolerance for a closed feature.

Begin at A and at
B save point
C begin feature
D end feature, save point
E begin feature
F end feature
A begin feature
B end feature

Figure 1 - Contour Clipping



Where begin point A and end point H are within the tolerance for a closed feature.

Begin at A and at
B save point
C B to C is beyond tolerance, save point
D begin feature
E end feature
F begin feature
G end feature
H get data at A
B begin feature
C end feature

Figure 2 - Contour Clipping

4. Symbol Library Improvement

A set of programs were provided under the Lineal to Raster Image Conversion Software to build the standard point symbol and alphabet libraries. These programs have been improved under this effort. The original program required each point symbol or character to be correctly digitized before the next sequential symbol or character is generated. Therefore, if a single character is in error, the entire alphabet must be redigitized.

The symbol library program was modified such that a code can be entered into the header of the point symbol or character to cause it to be deleted. The corrected character can be redigitized and added at the end of the file.

The original software also required that the digitizer had to be rezeroed for each symbol or character which was to be digitized. The new software performs a linear transformation on the data input, equating the best x and y coordinate contained in the symbol or character to zero. This transformation allows the zero point on the digitizer to be set only once. The new symbol library programs also guarantee closure in order that the area fill algorithm will properly image the symbol or character casing.

B. Format Conversion

1. Format Conversion Process Flow Optimization

The Preliminary Format Conversion Tests indicated a need for minimizing HIS-635 run times and hardware requirements. The initial Format Conversion Software required extensive CPU time and computer facilities. The first task was to reduce FCS run time requirements in order to reduce turn-around time. The reduction of turn-around time would provide increased opportunities to test new software.

The initial Format Conversion Software was designed such that a single module called each successive task . This process flow required the entire set of tasks to be run without interruption. Run times in the order of several hours were not unusual. The primary reason for the original structure was to verify all control cards in the initial task.

Data extracted from the control cards in the first task was stored in common and accessed by each successive task in order to avoid rereading of the input data.

The process flow was divided into eight individual tasks for use in the Raster Imaging Software. This was accomplished by placing the common data on a disk file and reading it into the common area of each task. The modified process flow not only increased turn-around time but also aided in the debugging process. It was now possible to debug a single module without rerunning all previous modules because the input files for each task were retained.

The Format Conversion Software uses several different data formats for each of the numerous points in a given run. Error detection and correction became a laborious task because of the shear volume of data which had to be converted to human readable format from the various interim digital formats. The conversion task was eliminated by the creation of a set of utility programs to print interim data files in a report format. These programs also provided the user with the ability to select only a subset of a data file to be plotted. This option provided a means to isolate problems which were related to only a localized area of the chart. These programs are documented in the Raster Imaging Software, Volume III, Program Documentation.

The Format Conversion Software process flow was altered a third time prior to the completion of the Raster Imaging Software effort. The FCS software was reduced to a three task process (Figure 3). This modification was implemented after the major debugging efforts had been completed. The three task structure eliminates four passes at the data files. This greatly reduces overall system run time (approximately 15%) while still maintaining multiple task flexibility.

The three task structure consists of a lineal phase which contains the ACS module, the LFEC module, and the INTL module. The ACS module reads the control cards and builds the common data file. Control is turned over to the LFEC module which calls the INTL module as a subroutine when

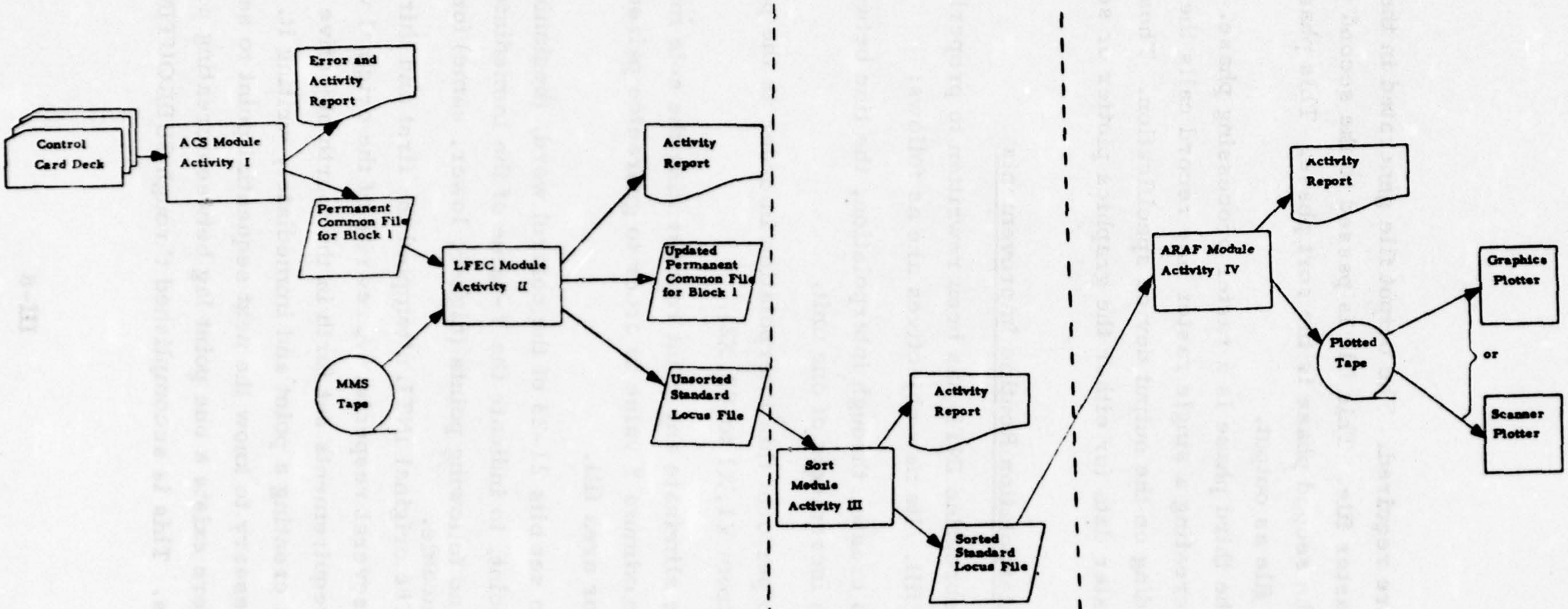


Figure 3 Format Conversion System Flow Chart

its services are required. The output file generated in the initial phase is an unsorted raster file. This file is passed to the second phase.

The second phase is the sort phase. This phase produces a sorted raster file as output.

The third phase is a raster processing phase. The ARAF module after creating a single raster line record calls the ROF or SCNPLT module depending on the output device specification. These modules reformat the raster data for either the graphics plotter or scanner/plotter, respectively.

2. Interpolation Routine Improvements

Subroutine INTL has been rewritten to properly prepare the data for area fill. Its new objectives are as follows:

- o to create, through interpolation, the line between two points in increments of one unit.
- o to perform this interpolation in order as the points are input (from Y1, X1 to Y2, X2).
- o to eliminate any point representing the sole minimum or maximum Y value in order to guarantee paired raster points for area fill.
- o to set bits 21-23 of the control word, (beginning at 0), for each point, to indicate the Y-value of the immediately preceeding and following points (higher, lower, same) for use by the ARAF module.

The original INTL attempted the first and third objectives, but failed in several respects. A rewrite of the original was necessary to meet the requirements set forth in the fourth objective above. The program was creating a point and immediately writing it. Since it now becomes necessary to know the next sequential point to satisfy the fourth objective, there exists a one point lag between creating points and outputting points. This is accomplished through SUBROUTINE SETTER.

Before a discussion of the methods utilized in fulfilling the above objectives, a brief overview of the program is given. It must be remembered that after being passed to the subroutine, there is a lag before the point is written. The first two points of the feature are read and interpolation is performed, determining the second point, which is saved. Interpolation continues along the line, writing points to the subroutine which sets the control word and outputs points to tape. When the end point of the line is reached, the next input point is read. The end point is then passed to the subroutine as a normal point or as a deleted point. If it is determined, upon creating the first point of the next line, that the deleted point must be output, a double pass is made through the subroutine maintaining the proper lag. Processing for each line continues in a similar manner until the end of feature is encountered. When the last input point equals the first input point, the feature is closed. This point is checked for delete characteristics, the point in lag is output, and if non-deleted, the final point is output. Processing then continues with the next feature. For open features, the point in lag, final point, and starting points are output. Processing continues until an end of file is encountered and all points have been output.

The four objectives were realized in the following manner:

a. Interpolation (first and second objectives)

Each line is regarded as going from the first input (Y1, X1) to the second input point (Y2, X2). For any line with a non-zero slope, interpolation is performed using the straight line equation. When the change in Y is greater than the change in X, Y1 is incremented by 1.0 (or -1.0 if Y1 is greater than Y2) and the equation is solved for X; otherwise X1 is incremented and the corresponding Y value determined. This is repeated until the rounded integer of the point created is one less (or greater if the increment is -1.0) than the rounded integer value of Y2 or X2, respectively. In the case of a horizontal or vertical line, Y1 or X1 is incremented by ±1.0, respectively and the process is repeated a similar number of times. This insures that points are created for each whole unit (integer) from Y1, X1 to Y2, X2.

b. Deleted Points (third objective)

The point representing a change in Y direction must be eliminated providing that:

- o it is neither end point of a horizontal line and,
- o no point was created, on either the line ending with or beginning with that point, which had the same Y value. This implies that the next input point is read before the delete condition is determined.

c. Control Word (fourth objective)

Setting bits 21-23 of the control word requires that the preceeding and following points are known. This is accomplished through the lag in subroutine SETTER. At the beginning of each feature, the first input Y is stored as LASTY and the first created point is stored as SECNDY, SECNDX and NOWY, NOWX. For all other points, the relative value of the preceeding Y is determined by comparing LASTY and NOWY. Similarly, the relative value of the following Y is determined by comparing NOWY and NEXTY (NEXTY being the most recently created point). The proper value is then masked into the control word and NOWY, NOWX is output. NOWY is stored in LASTY and NEXTY/NEXTX are stored in NOWY, NOWX before control is returned to the calling routine. This subroutine is called as each point is to be output. For closed features, the SECNDY, SECNDX are used to determine the control word for the final point to be output. For open features, the point in lag is output with the proper control word, but the control word of the beginning and ending points remains unchanged.

The relatively simple process of lineal interpolation increases in complexity when considering the following requirements.

- o greatest possible accuracy (implying that no extraneous points should be output).
- o least possible time.
- o information provided to area fill

- lower, same, higher Y properties or preceeding and following points.
- a minimum of 2 points for each Y in any single closed feature.

The following graph illustrates the output from the latest version of INTL (Figure 4).

3. Sort Optimization

The sorting of the raster records into X within Y order for use by the area fill algorithm and ultimately by the plotter is the most time consuming process of the Raster Imaging Software. A number of technqiues were tried to optimize this process including increasing the number of buffers and the number of sort work files. None of these techniques made significant improvements. PRC did, however, discover that packing data into records of reduced size increased speed by about 10%. PRC also performed experiments to compare the speed of the HIS-635 sort with the U-1108 sort speed. The HIS-635 is currently being used for system development. The ultimate user, however, will be the U-1108. A 50% speed increase was indicated by the use of the 1108 over the HIS-635 in sorting the control data set. This speed increase was significant enough to indicate that further improvement to the HIS-635 sort would be of little consequence when a production system is implemented on the U-1108 system.

4. Area Fill Program

The Area Fill Program performs raster imaging and is the most complex program of the entire system. The Raster Imaging Software Area Fill program uses a totally different approach from the original Lineal to Raster Image Conversion Software Area Fill Algorithm.

The Area Fill Program performs all raster imaging functions including 1) area fill, 2) areal line weight, 3) priority masking, 4) fill between, 5) screen and hatchure fill, 6) repeated symbology fill, and 7) cased road intersection masking. All of the above functions are inter-related, and therefore, are performed simultaneously.

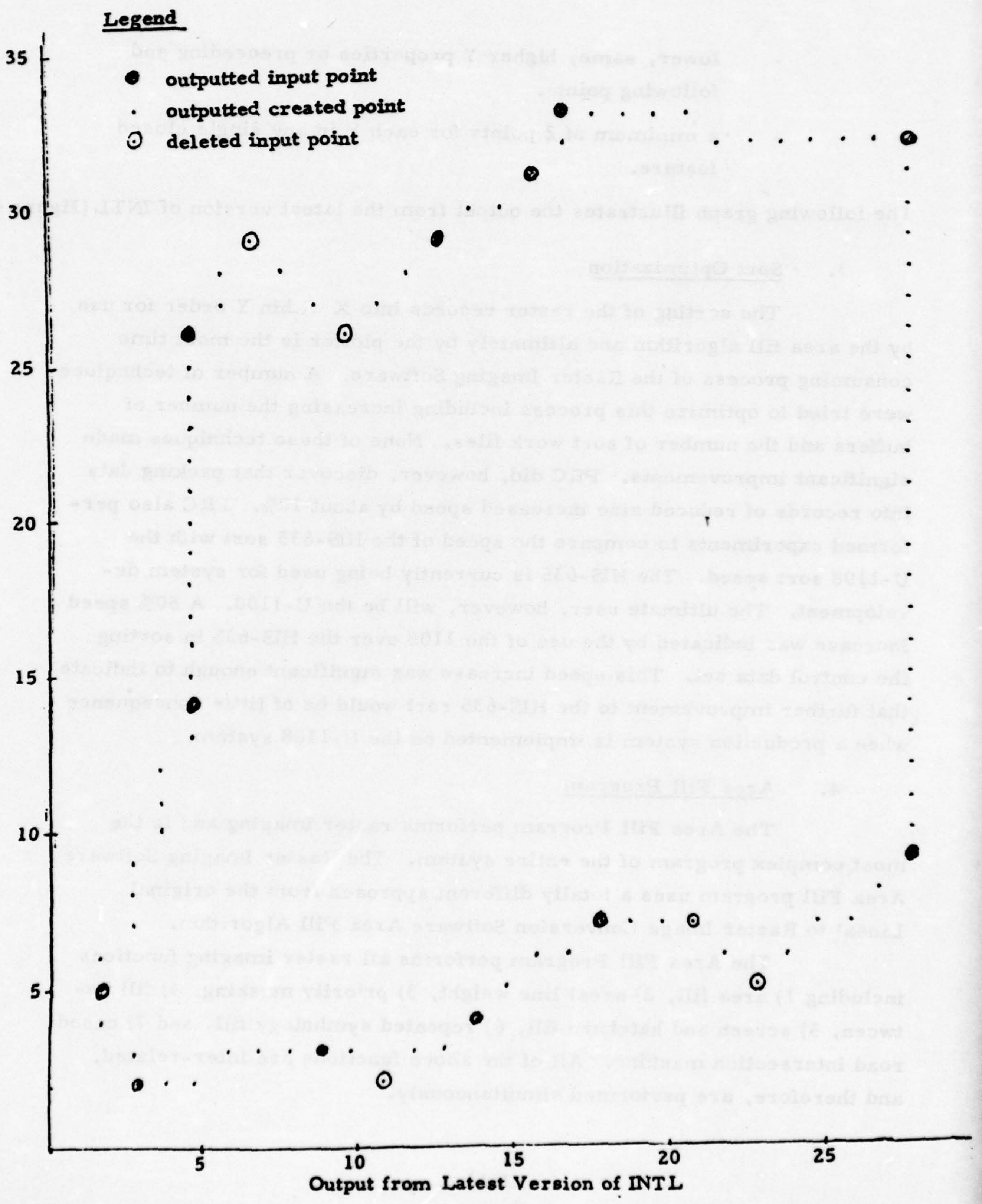

Legend
outputted input point
outputted created point
deleted input point
35
30
25
20
15
10
5
5
10
15
20
25
Output from Latest Version of INTL

Figure 4

The current Area Fill algorithm uses a push-down stack combined with a linked list to evaluate each raster line. The entire raster line is scanned to produce the two lists. The lists are then processed. The points which delimit the areas to be filled are paired. The priority scheme then determines the given color or gray shade to be output at any given point on the raster line.

All the raster imaging functions listed above are defined as area fills. The areal line weights are delineated by closed line casings generated by the lineal processing prior to the sort. The cased roads are generated by defining two areal lines for each road. The first areal line width is specified as the distance between the outer edges of the casing. The second areal line weight is specified as the distance between the inner edges of the casing. The first line is given a lower priority than the second line. The second line is specified to be blank filled. The two superimposed lines produce the desired case when area fill is performed on each line. All other raster functions are related to features which are closed in their center line format.

Screens, hatchures, and repeated symbology fills are produced in raster format through subroutine calls when an area to be filled is determined by the Area Fill Program. These subroutines define the raster patterns which produce the desired fill and specify the starting point within the pattern. The input to the subroutine is the (X, Y) pair defining the start point of fill. The subroutine returns the pattern data unique to the given (X, Y) pair. In this manner the subroutine relieves the calling program from keeping track of the position of a given raster line relative to overall pattern.

IV. FINAL TEST RESULTS

The final phase of the Raster Imaging Software test effort was to evaluate the enhanced raster conversion process. The test procedures were defined as follows.

A. Test Data Base

The upper right corner defined by the coordinates (N 111° 50' 30" W 33° 22' 30") and (N 112°, W 33° 30') of the 1952 map of Tempe, Arizona at a scale of 1:2400 was used as the final test data. This sector contains a relatively even distribution of graphic data and a representative cross section of cardographic data.

B. Format Conversion Tests

Perform the Format Conversion runs by coding the control cards to extract the following feature classes with the specified function and priority each with an output resolution of 1000 lines per inch.

TEST 1	function	priority
SL Permenant Drains	line center	10
SL Non Permenant Drains	line center	5
Lake	area fill	20
Aqueduct	line center	11
Levee	line center	12

TEST 2

	function	priority
Canals	line center	10
Sand Area	screen 90^o	15
Sewage Disposal	swamp grass pattern	40
Fish pond	area fill	15
Filter Bed	rice pattie pattern	40
Race Track	line center	20
Orchard	area fill	25
Railroads	line center	20
Railroad Yard	line center	21
Railroad Siding	line center	22
Railroad Main Line	line center	23
Building	Screen 45^o	24
Landmark Features	area fill	25
River Lines	line center	10
Pipe line	line center	10
Outlined Landmark	Screen 135^o	30
City Outline	area fill	50
Tank	line center	10
Race Track	line center	30

TEST 3

	function	priority
Thruway	case	30
(Inner area)		40
Trail	line center	10
Medium Duty	area fill	20
Streets in City	case	30
(Inner area)		40

TEST 4

	function	priority
Light duty roads	line center	10
Light duty roads	line center	20

C. Test Results and Statistics

All of the Format Conversion tests ran to normal termination on the Honeywell 6180 (Note: Honeywell 6180 replaced Honeywell 635/45 prior to final testing).

In order to review the results of the tests, the resulting plotter tapes were read into the Raster Edit System developed under the Raster Lineal Editing Software contract. The data was then displayed on the Textronix T-4002A display terminal. This route was taken to review the test results because the Graphics Plotter was inoperable.

The results of the Format Conversion testing proved that each function is operational; the statistical timings are in Figures 1 - 4. The enhanced Format Conversion Software delivered to RADC under contract No. F30602-74-C-0345 is capable of performing areal line weights, priority masking, area fill, fill between, screen and hatchure fill and repeated symbol fill accurately.

The comparison of the timings on the original Raster Imaging Software with the current system is difficult to evaluate. The Preliminary Format Conversion Test timings were incomplete since not all the tests ran to normal termination (sec Test Plans/Procedures Report). An analysis of the timings obtained from the Preliminary and Final Tests was accomplished by using the following formula:

let	PF	=	Points Final (Number of points manipulated for final test)
	PP	=	Points Preliminary (Number of points manipulated for initial test)
	TF	=	Time Final Expected (expected run time based on time/point of the initial test)
	TP	=	Time Preliminary (run time of preliminary test)

A proportion can be set up such that

$$\frac{PF}{PP} = \frac{TF}{TP}$$

solve for TF

$$PF*TP = PP*TF$$

$$PF * \frac{TP}{PP} = TF$$

Note: $\frac{TP}{PP}$ is the time per point on the initial test.

	Processor Time (sec)	Percentage Module/Total Time/Time	Points Input	Points Output	Points Deleted	Points Added
Total Job	276.12	100.00				
ACS	1.44	.54				
LFEC	59.76	21.6	15274	96885		81611
SORT	106.56	38.6	96885	96885		
ARAF	108.36	39.26	96882	96748	137	

STATISTICAL TIMINGS TEST 1

Figure 1

	Processor Time (sec)	Percentage Module/Total Time/Time	Points Input	Points Output	Points Deleted	Points Added
Total Job	529.56	100.00				
ACS	1.44	.27				
LFEC	115.2	21.76	13123	195918		182794
SORT	218.88	41.33	195918	195869	49	
ARAF	194.04	36.64	195863	208725	5	12862

STATISTICAL TIMING TEST 2

Figure 2

	Processor Time (sec)	Percentage Module/Total Time/Time	Points Input	Points Output	Points Deleted	Points Added
Total Job	534.24	100.00				
ACS	1.44	.27				
LFEC	118.08	22.10	1036	197064		196028
SORT	217.44	40.70	197064	196691	373	
ARAF	197.28	36.93	191717	212284	4974	20567

STATISTICAL TIMINGS TEST 3

Figure 3

	Processor Time (sec)	Percentage Module/Total Time/Time	Points Input	Points Output	Points Deleted	Points Added
Total Job	500.04	100.00				
ACS	1.44	.29				
LFEC	108.36	21.71	10017	193400		183383
SORT	212.40	42.47	193400	193399	1	
ARAF	177.84	35.53	193397	193397	2	

STATISTICAL TIMINGS TEST 4

Figure 4

The first set of expected final timings (Figure 5) were calculated using the total number of points processed in the Final Tests and the total number of points processed in the Preliminary Test (estimated since the ARAF program had an error and did not report the number of points processed). This showed that in every case except one the true final run time was less than the expected final run time by an average of 55%.

The second set of expected final timings (Figure 6) were calculated using the total of the number of points input plus number of points output in the Final Test and the total of the number of points input plus the number of points output in the Preliminary test. The reason for using the second method was due to the fact that the number of points input to the FCS increased when passed through the casing and INTL algorithms. This increase in the number of points was reflected in the timings of the sort and area fill modules. Since the first method was required to use estimated timings because of a lack of a report in the area fill module for the Preliminary Test, this second method was used. The total number of points input and the total number of points output was known in both the Preliminary and Final Tests. This showed that in all cases the true final run time was less than the expected final run time by an average of 255%.

This evidence has been presented, but is difficult to evaluate accurately. It does not reflect a total overview since the results of the Preliminary Tests were obtained from software which did not work completely or have all the capabilities of the software used for Final Testing. The Raster Imaging Software being delivered to RADC has been greatly improved from the original software and has many more capabilities that we can only conclude that we are providing the user with a faster, more useful system.

Test Number	Preliminary Total Points Processed	Final Total Points Processed	Preliminary Total Time (sec)	Final Total Time (sec)	Expected Final Time ** Based on tests 1	3	4	5
1	*2922000	449559	1764.36	276.12	271	355	476	408
2	ABORTED	1005416	ABORTED	529.56	607	794	1065	913
3	*1032600	995856	815.76	534.24	601	787	1055	904
4	*276600	977010	293.04	500.04	590	772	1035	866
5	*371400		336.96					
6	ABORTED		ABORTED					

* estimated since the ARAF program had an error and did not report the number of points processed.

** Since timings dependent on FCS options specified at run time, Expected Final Run Times have been calculated on the basis of time per point on each of the successful preliminary tests

Expected Final Timings
Method No. 1

Figure 5

Test Number	Preliminary Total Points Input & Output	Final Total Points Input & Output	Preliminary Total Time (sec)	Final Total Time (sec)	Expected Final Time ** Based on tests 1	3	4	5
1	337336	112022	1764.36	276.12	586	813	1673	852
2	ABORTED	221848	ABORTED	529.56	1160	1610	3312	1685
3	112371	213320	815.76	534.24	1116	1549	3185	1620
4	19625	203414	293.04	500.04	1064	1476	3037	1546
5	44352		336.96					
6	ABORTED		ABORTED					

** Since timings dependent on FCS options specified at run time, Expected Final Run Times have been calculated on the basis of time per point on each of the successful preliminary tests

Expected Final Timings
Method No. 2

Figure 6

V. CONCLUSIONS AND RECOMMENDATIONS

The Raster Imaging Software effort has demonstrated two basic capabilities:

- o Point symbols and alphanumeric text strings may be positioned and generated through the use of graphic display devices
- o Raster imaging of center line files is a practical means of producing final color separations.

A. Point Symbol, Alphanumeric Character, and Contour Label Placement

The symbol and character positioning techniques developed under the Experimental Compilation Console should be modified for production usage.

A display device having a higher resolution and smaller spot size will be required in the production environment. The Tektronix 611 CRT has a 15 mil spot size which causes sufficient blossoming to completely distort symbols displayed at 1X. Scaling to 2X produces distinguishable symbols, but the increase in scale cause produces the viewed area by a factor of 4. The 10 mil resolution of the Tektronix 611 causes significant distortion in the point symbols due to their relative minute size.

The use of a base line to define the graphic symbols position should be eliminated and replaced with a fully interactive procedure. The graphic symbol position and orientation could be defined with an input device that detects rotation as well as (x, y) movement. The appropriately plotted symbol should immediately reflect the input device position. This level of interactivity will require a refresh CRT due to the constant updating requirement.

A new data format which is oriented toward storing multiple pen strokes which are logically related should be adopted for the storage of point symbols and alphanumeric character strings. The MMS format is *not easily adapted to the above requirements.*

The preceding comments are also applicable to the contour labeling

techniques. However, the contour labels require a somewhat more complex data structure. The contour record should contain data which specifies the position of the label and the type style. The contour label data should only be separated from the associated contour when it is logically unrelated to the contour (e.g. names overlay production). This approach will eliminate the need for maintaining a separate contour label record.

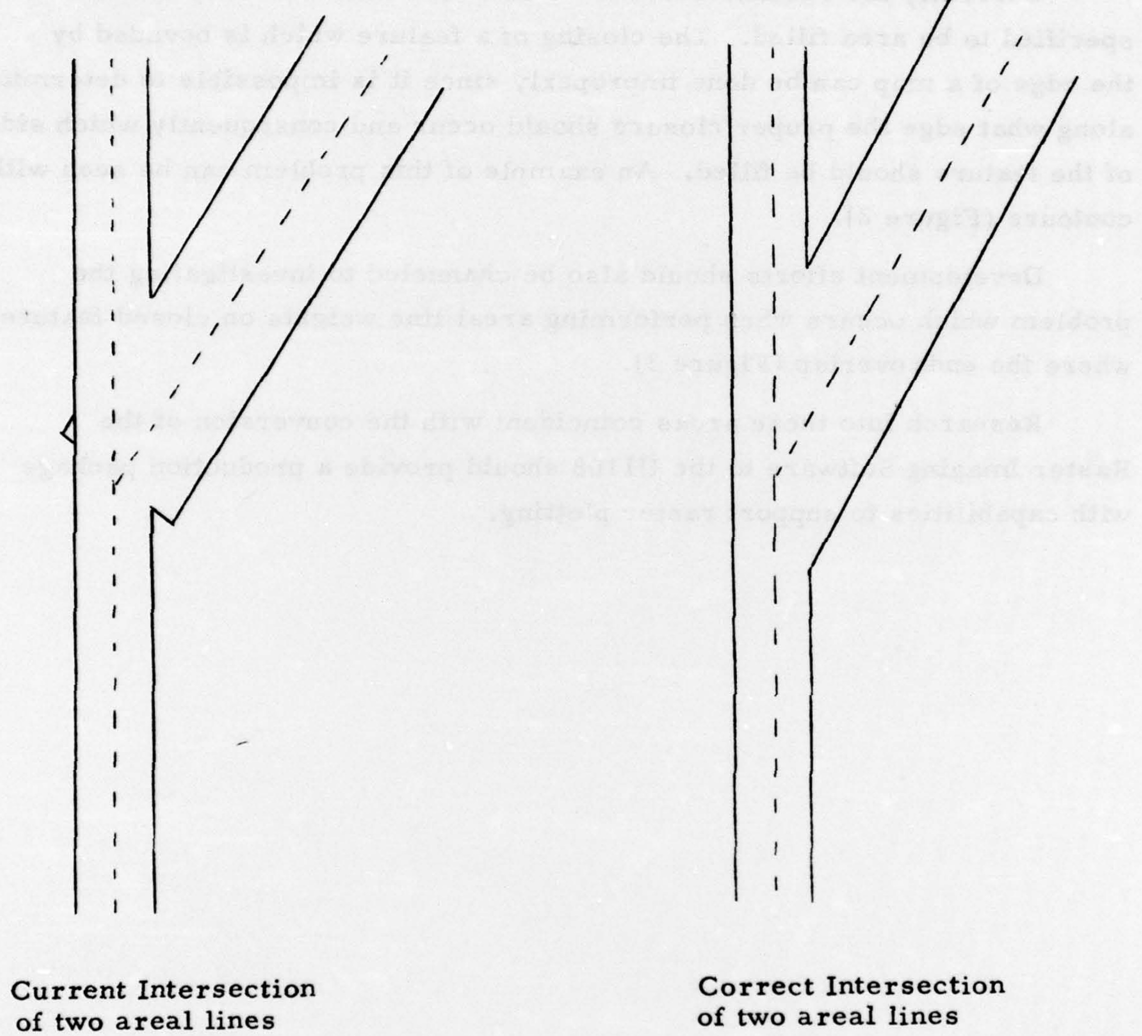
PRC recommends that the techniques explored for point symbol, alphanumeric character, and contour label placement be used as a basis for development of an advanced Automatic Type Placement System (ATPS). Our experimentation has indicated that the use of interactive displays are useful in the production of names overlay. The design of a prototype ATPS should be undertaken improving the experimental software and integrating it with a high precision display device.

B. <u>Format Conversion</u>

All raster imaging functions are ready for generation of a production prototype system. Further testing of the software using varied sources for input should be undertaken. The purpose of this testing would be to further verify the algorithms prior to conversion for the U-1108. Once the algorithms have been throughly checked, conversion to U-1108 Fortran or assembly language should be commenced.

Future development efforts should be directed toward creation of software to generate the proper intersection of two areal lines crossing at acute angles (Figure 1). Occurance of these types of intersections are relatively infrequent but with the current system these intersections must be retouched.

The repeated pattern fill is another area which will require additional research. A number of the patterns are random and have no determinable repeat. These types of patterns do not lend themselves to automation. DMA should consider modifying patterns such as to make them more amenable to area fill automation.



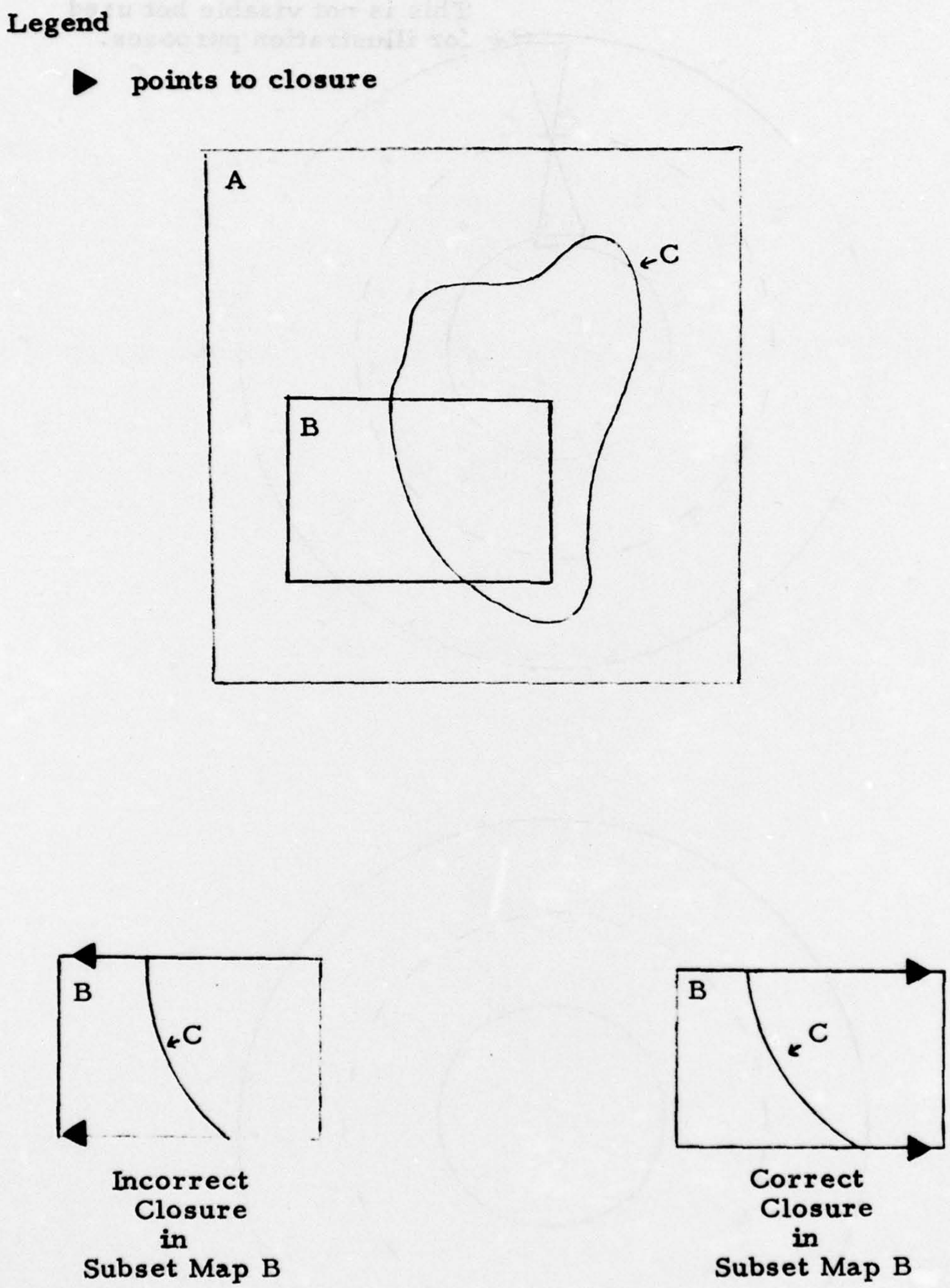
Intersection of Two Areal Lines at Acute Angles

Figure 1

Currently the Format Conversion Software will close any feature specified to be area filled. The closing of a feature which is bounded by the edge of a map can be done improperly since it is impossible to determine along what edge the proper closure should occur and consequently which side of the feature should be filled. An example of this problem can be seen with contours (Figure 2).

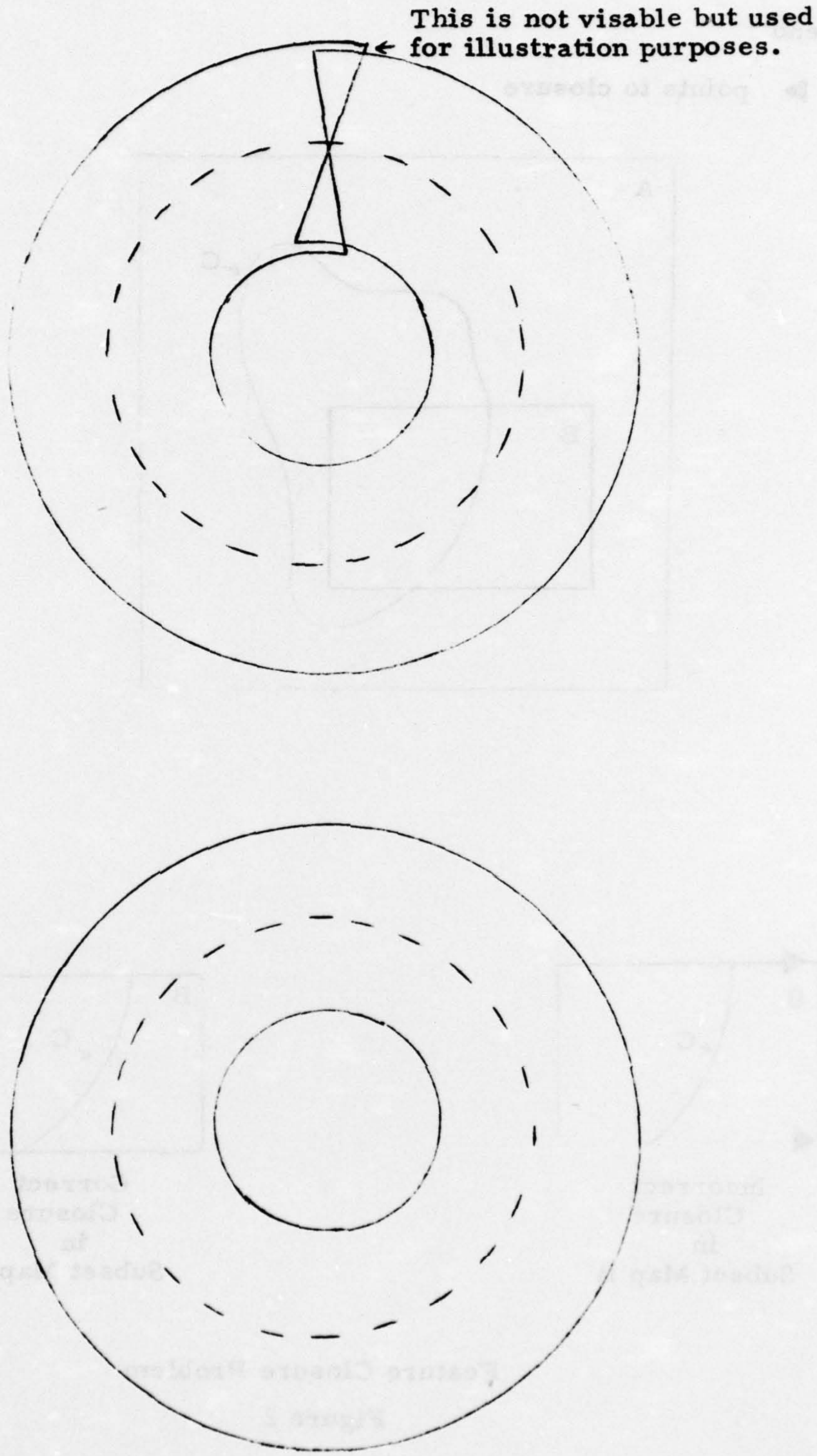
Development efforts should also be channeled to investigating the problem which occurs when performing areal line weights on closed feature where the ends overlap (Figure 3).

Research into these areas coincident with the conversion of the Raster Imaging Software to the U1108 should provide a production package with capabilities to support raster plotting.



Feature Closure Problem

Figure 2



Areal Line Weights on Closed Feature
Figure 3

METRIC SYSTEM

BASE UNITS:

Quantity	Unit	SI Symbol	Formula
length	metre	m	...
mass	kilogram	kg	...
time	second	s	...
electric current	ampere	A	...
thermodynamic temperature	kelvin	K	...
amount of substance	mole	mol	...
luminous intensity	candela	cd	...

SUPPLEMENTARY UNITS:

Quantity	Unit	SI Symbol	Formula
plane angle	radian	rad	...
solid angle	steradian	sr	...

DERIVED UNITS:

Quantity	Unit	SI Symbol	Formula
Acceleration	metre per second squared	...	m/s
activity (of a radioactive source)	disintegration per second	...	(disintegration)/s
angular acceleration	radian per second squared	...	rad/s
angular velocity	radian per second	...	rad/s
area	square metre	...	m
density	kilogram per cubic metre	...	kg/m
electric capacitance	farad	F	A·s/V
electrical conductance	siemens	S	A/V
electric field strength	volt per metre	...	V/m
electric inductance	henry	H	V·s/A
electric potential difference	volt	V	W/A
electric resistance	ohm		V/A
electromotive force	volt	V	W/A
energy	joule	J	N·m
entropy	joule per kelvin	...	J/K
force	newton	N	kg·m/s
frequency	hertz	Hz	(cycle)/s
illuminance	lux	lx	lm/m
luminance	candela per square metre	...	cd/m
luminous flux	lumen	lm	cd·sr
magnetic field strength	ampere per metre	...	A/m
magnetic flux	weber	Wb	V·s
magnetic flux density	tesla	T	Wb/m
magnetomotive force	ampere	A	...
power	watt	W	J/s
pressure	pascal	Pa	N/m
quantity of electricity	coulomb	C	A·s
quantity of heat	joule	J	N·m
radiant intensity	watt per steradian	...	W/sr
specific heat	joule per kilogram-kelvin	...	J/kg·K
stress	pascal	Pa	N/m
thermal conductivity	watt per metre-kelvin	...	W/m·K
velocity	metre per second	...	m/s
viscosity, dynamic	pascal-second	...	Pa·s
viscosity, kinematic	square metre per second	...	m/s
voltage	volt	V	W/A
volume	cubic metre	...	m
wavenumber	reciprocal metre	...	(wave)/m
work	joule	J	N·m

SI PREFIXES:

Multiplication Factors	Prefix	SI Symbol
1 000 000 000 000 = 10^{12}	tera	T
1 000 000 000 = 10^{9}	giga	G
1 000 000 = 10^{6}	mega	M
1 000 = 10^{3}	kilo	k
100 = 10^{2}	hecto*	h
10 = 10^{1}	deka*	da
0.1 = 10^{-1}	deci*	d
0.01 = 10^{-2}	centi*	c
0.001 = 10^{-3}	milli	m
0.000 001 = 10^{-6}	micro	μ
0.000 000 001 = 10^{-9}	nano	n
0.000 000 000 001 = 10^{-12}	pico	p
0.000 000 000 000 001 = 10^{-15}	femto	f
0.000 000 000 000 000 001 = 10^{-18}	atto	a

* *To be avoided where possible.*

*U.S. GOVERNMENT PRINTING OFFICE: 1977-714-025/187

MISSION
of
Rome Air Development Center

RADC plans and conducts research, exploratory and advanced development programs in command, control, and communications (C^3) activities, and in the C^3 areas of information sciences and intelligence. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.

AMERICAN REVOLUTION BICENTENNIAL 1776-1976